MilkyTracker Manual v1.0.1

Hello and welcome to MilkyTracker, an open source multi-platform Fasttracker II compatible music tracker program. This document holds a lot of valuable information about the tracker but it's not a tracking manual. If you want to learn more about tracking and how it's done, the Internet is your friend. We host some resources on MilkyTracker.titandemo.org as well.

Disclaimer:
MilkyTracker is under development so use it at your own risk. The team is not responsible for any loss of data and/or hardware damage caused by MilkyTracker.

Now, with the formalities taken care of, let's have a look at the…

Table of Contents

1. Supported platforms

MilkyTracker currently runs on the following platforms:

- AmigaOS 4
- AROS
- *BSD
- GNU/Linux
- Haiku
- Mac OS X
- Microsoft Windows 9x/NT/Me/200x/XP/Vista
- Microsoft Windows CE 3.0+/Mobile

It has also been known to run on:

- Xbox (GNU/Linux)
- Solaris 9 & 10
- Sony PlayStation Portable

You can run MilkyTracker on other platforms by compiling from source.

2. Overview

MilkyTracker is an open source, multi-platform music application; more specifically part of the tracker family. It attempts to recreate the module replay and user experience of the popular DOS application Fasttracker II, with special playback modes available for improved Amiga ProTracker 2.x/3.x compatibility. MilkyTracker is not "another Windows tracker", which should already be made obvious by the plethora of supported desktop and portable platforms. In fact it started as a project to bring tracking to the Pocket PC. When this milestone was reached, the next one was creating a truly FT2 compatible tracker for portable as well as modern desktop platforms.

3. Features

- Fasttracker II-like, custom graphical user interface with context menus
- Supported on multiple platforms including portable devices
- Very accurate .XM replay compared to FT2
- ProTracker 2/3 playback modes for playing and editing .MOD files
- Various resampler options including emulated Amiga 500/1200 sound output
- Choose between a modern and a true-to-FT2 editing scheme / keyboard layout
- Tabbed user interface for opening and playing up to 32 modules simultaneously and for exchanging data between them
- Over 30 imported module formats
- Basic archive support for loading zipped, powerpacked and UMX modules directly
- Rendering songs/patterns to disk (.WAV) or directly to sample
- Powerful sample editor featuring waveform generators
- In-depth instrument editor featuring envelope zooming and scaling and support for copying and swapping instruments and samples across tabs
- Copy/swap dialog for instrument management
- Undo/redo in pattern/sample/instrument editor
- Low latency audio driver support
- MIDI In support
- Module optimizer
- Internal file browser option
- Various font sizes for improved visibility of pattern data
- Prospective pattern view option
- Playing and editing simultaneously
- Live mode for seamless pattern changes

Resamplers

MilkyTracker offers various resampling options for module playback, rendering and sample processing. These are:

- No interpolation
- Linear interpolation
- Cubic Lagrange
- Cubic Spline
- Fast Sinc (window size 16, fixed point integer, sinc lookup table)
- Precise Sinc (window size 128, double floating point)
- Amiga 500
- Amiga 500 LED
- Amiga 1200
- Amiga 1200 LED

While the choice of resampler is a matter of personal taste, you should keep in mind that **Linear interpolation** represents the highest quality option available in Fasttracker II so that's what the majority of .XM files were probably made (to be played) with. Many chiptunes will however sound very muffled with interpolation because of their short samples and therefore relatively greater impact of interpolation. The Amiga modes are meant to be used with 4 channel .MODs only. **Precise Sinc** is a CPU killer - great for resampling in the sample editor but don't expect hot real-time performance.

Tabs

MilkyTracker enables you to open and play up to 32 modules simultaneously and to exchange data between them. Initially, tabs are invisible but can be activated with keyboard shortcuts described below. There are some configurable options for tabs as well, like automatically opening modules in new tabs and background stopping behavior control. You can choose to never stop playback on background tabs, or to automatically stop on tab change or stop when playback on another tab is started. Playback can also resume upon returning to a tab.

4. Imported and exported file formats

I. Modules

MilkyTracker can import a wide range of tracker module formats but since Milky is a FT2 clone, modules are replayed in an FT2 environment which means not all features of different formats are supported. MilkyTracker also has basic archive support, so it's possible to load zipped, powerpacked and UMX modules directly.

Import:

.669	669 Composer/Unis669 (PC)
.AMF	Asylum Music Format ("Crusader" in-game music) (PC) Digital Sound and Music Interface (DSMI) library (PC)
.AMS	Extreme Tracker (PC) Velvet Studio (PC)
.CBA	Chuck Biscuits+Black Artist module format (PC)
.DBM	DigiBooster Pro (Amiga)
.DIGI	Digibooster 1.0-1.7 (Amiga)
.DSM	Digisound Interface Kit (DSIK) library (PC) Dynamic Studio (PC)
.DTM	Digital Tracker (Atari) DigiTrekker 3.0 (PC)
.FAR	Farandole Composer (PC)
.GDM	General Digimusic (PC)
.GMC	Game Music Creator (Amiga)
.IMF	Imago Orpheus (PC)
.IT	Impulse Tracker (PC)
.MDL	DigiTrakker 1.0-3.0 (PC)
.MOD	Sound-/ProTracker and variants (Amiga & PC)
.MTM	MultiTracker (PC)
.MXM	Cubic Tiny XM (PC)
.OKT	Oktalyzer (Amiga)
.PLM	DisorderTracker II (PC)
.PSM	Epic MegaGames MASI (PC)
.PTM	PolyTracker (PC)
.S3M	Scream Tracker 3.0 (PC)
.SFX	SoundFX (Amiga)
.STM	Scream Tracker 2.0 (PC)
.ULT	UltraTracker (PC)
.UNI	MikMod (PC)
.XM	Fasttracker II (PC)

MilkyTracker's song export options are the same as Fasttracker II's. MilkyTracker also features ProTracker 2.x and 3.x playback modes for .MODs.

Export:

.MOD	ProTracker boundaries (including 64kb max sample length), although can save 2–32 channels
.WAV	Microsoft/IBM PCM Waveform audio rendering
.XM	Fasttracker II compatible, not as common as one might think

II. Samples

Milky can load practically anything as RAW PCM audio samples; one of FT2's famous features.

Import:

.8SVX / **.IFF**	Compressed/uncompressed Interchange File Format
.AIF / **.AIFF**	Apple Audio Interchange File Format
.WAV	Microsoft/IBM uncompressed PCM Waveform audio
.*	RAW PCM audio

Export:

.IFF	Uncompressed Interchange File Format
.WAV	Microsoft/IBM uncompressed PCM Waveform audio

III. Instruments

MilkyTracker can load and save FT2's eXtended Instrument (**.XI**) format and additionally import Gravis Ultrasound GF1 Patch (**.PAT**) files.

IV. Patterns and Tracks

MilkyTracker handles FT2's eXtended Pattern (**.XP**) and eXtended Track (**.XT**) files with full compatibility.

5. Keyboard shortcuts

By user request, MilkyTracker features two edit modes. You can switch between these in the Config screen (Misc. tab). To learn about the differences and which might better suit you, read the appropriate sections below. There are a couple of shortcuts that are the same for both modes so let's clear those out of the way first:

Please note that under Mac OS X the Command key is used instead of the Ctrl key.

Alt-Enter	Switch between full screen and windowed display (Windows & SDL)
Shift-Command-F	Switch between full screen and windowed display (OS X)
Shift-M	Mute current channel
Ctrl-Shift-M	Invert muting
Shift-U	Un-mute all
Ctrl-Shift-T	Open a new tab
Ctrl-Shift-W	Close current tab
Ctrl-Shift-Left	Select previous tab
Ctrl-Shift-Right	Select next tab
Ctrl-=	Increment instrument number of all notes in the current selection
Ctrl--	Decrement instrument number of all notes in the current selection
Ctrl-Shift-=	Increment instrument number of all notes in the current track under the cursor
Ctrl-Shift--	Decrement instrument number of all notes in the current track under the cursor

I. MilkyTracker edit mode

The MilkyTracker mode basically is a bit more "modern" because you can focus on different parts (e.g. Pattern Editor, Instrument listbox, Sample listbox etc.) and when you're pressing keys, they're routed to the focused control. Keyboard shortcuts are also more standard; you can select by pressing the SHIFT key and navigating with the cursor keys, cut, copy & paste by using Ctrl-X/C/V etc. Users who are new to tracking will probably find this a bit more intuitive.

Section switching:

Ctrl-Alt-

A	Advanced edit
C	Configuration
D	Disk operations
I	Instrument editor
R	Disk recorder
S	Sample editor
T	Transpose
X	Main screen
Z	Toggle scopes

Global:

2, 3, 5, 6… **Q, W, E, R…** **S, D, F, G…** **Z, X, C, V…**	Play / insert notes (depending on whether edit mode is on)
F1…F8 **Ctrl-Shift-1…8**	Select octave
Space	Toggle pattern editor focus (edit mode on/off)
Enter	Play song from current order
Ctrl-Enter	Play current pattern from beginning

Shift-Enter	Play current pattern from cursor position
Shift-F9	Play current pattern from beginning (same as Ctrl-Enter)
Shift-F10	Play current pattern from position after the first quarter of the pattern length
Shift-F11	Play current pattern from position after the second quarter of the pattern length
Shift-F12	Play current pattern from position after the third quarter of the pattern length
Alt-Space	Play song from current row (stop and return when keys are released)
Shift-Space	Play row by row
Esc	Stop
Ctrl-F	Toggle song follow
Ctrl-P	Toggle prospective pattern view
Ctrl-W	Toggle pattern wrapping
Ctrl-L	Toggle pattern change behavior (live mode)
Ctrl-O	Load song
Ctrl-S	Save song
Ctrl-Shift-S	Save song as…
Ctrl-Q	Exit program
Alt-F4	

Pattern Editor:

Cursor keys	Move around
Tab	Jump to next channel
PageUp	Jump 16 rows up
PageDown	Jump 16 rows down
Home	Jump to first row
End	Jump to last row
F9	Jump to beginning of the pattern
F10	Jump to position ¼ through the pattern
F11	Jump to position halfway through the pattern
F12	Jump to position ¾ through the pattern
Ctrl-Z	Undo
Ctrl-Y	Redo
Shift-Cursor keys	Select block
Shift-Alt-Cursor keys	Extend block
Ctrl-A	Select entire pattern
Ctrl-X	Cut
Ctrl-C	Copy
Ctrl-V	Paste
Ctrl-I	Interpolate values
Delete	Delete note/instrument/volume/effect/parameter
Shift-Del	Delete note, volume and effect at cursor
Ctrl-Del	Delete volume and effect at cursor
Alt-Delete	Delete effect at cursor
Insert	Insert space on current track at cursor position
Shift-Insert	Insert row at cursor position
Backspace	Delete previous note
Shift-Backspace	Delete previous row
The key right of LShift	Enter key-off
The key below Esc	Enter key-off (Windows only)
1	Enter key-off (OS X only)
Alt-Minus	Increase Add value
or Alt-Plus	Decrease Add value

Transpose:

Alt-F7	Transpose current instrument in block down
Alt-F8	Transpose current instrument in block up
Shift-F7	Transpose current instrument in track down
Shift-F8	Transpose current instrument in track up
Ctrl-F7	Transpose current instrument in pattern down
Ctrl-F8	Transpose current instrument in pattern up
Alt-F1	Transpose all instruments in block down
Alt-F2	Transpose all instruments in block up
Shift-F1	Transpose all instruments in track down
Shift-F2	Transpose all instruments in track up

Ctrl-F1	Transpose all instruments in pattern down
Ctrl-F2	Transpose all instruments in pattern up

Sample Editor:

Shift & drag	Quick draw
Ctrl & drag	Resize selection
Alt & drag	Move selection or loop range

II. Fasttracker II edit mode

The FT2 edit mode is for the die-hard FT2 users and probably isn't very intuitive to beginners. **Please note** that not all FT2 shortcuts are implemented yet and some may differ for various technical reasons. **Also note** that this edit mode may not be optimal on Pocket PC because of the limitations of some input devices.

Section switching:

Ctrl-	
A	Advanced edit
C	Configuration
D	Disk operations
I	Instrument editor
R	Disk recorder
S	Sample editor
T	Transpose
X	Main screen
Z	Toggle scopes

Global:

2, 3, 5, 6… **Q, W, E, R…** **S, D, F, G…** **Z, X, C, V…**	Play / insert notes (depending on whether edit mode is on)
F1…F8	Select octave
Right Ctrl	Play song from current order
Enter	Play song from current order
Right Alt	Play current pattern from beginning (Windows &SDL)
Ctrl-Enter	Play current pattern from beginning
Shift-Enter	Play current pattern from current row
Shift-F9	Play current pattern from beginning (same as Ctrl-Enter/Right Alt)
Shift-F10	Play current pattern from position after the first quarter of the pattern length
Shift-F11	Play current pattern from position after the second quarter of the pattern length
Shift-F12	Play current pattern from position after the third quarter of the pattern length
Alt-Space	Play song from current row (stop and return when keys are released)
Shift-Space	Play row by row
Space	Stop / Edit
Shift-Left	Increase song position
Shift-Right	Decrease song position
Ctrl-Left	Increase current pattern number
Ctrl-Right	Decrease current pattern number
Ctrl-F9	Delete current order position
Ctrl-F10	Insert new order position
Ctrl-F11	Decrease current order pattern number
Ctrl-F12	Increase current order pattern number
Key below ESC (ANSI: Alt-Minus)*	Increase Add value
Shift-key below ESC (ANSI: Alt-Plus)*	Decrease Add value
Ctrl-F	Toggle song follow
Ctrl-P	Toggle prospective pattern view
Ctrl-W	Toggle pattern wrapping
Ctrl-L	Toggle pattern change behavior (live mode)

Shift-Ctrl-L	Load song
Shift-R	Toggle record mode
Shift-Ctrl-S	Save song
Esc	Exit program

* **Please note** in this table, "Key under esc" refers to the tilde / tick key, section symbol / plusminus key or the ring accent / circumflex key depending on your ISO keyboard, but does not exist on ANSI layouts. See: this issue for details

Pattern editor:

Cursor keys	Move around
PageUp	Jump 16 rows up
PageDown	Jump 16 rows down
Home	Jump to first row
End	Jump to last row
Tab	Jump to next track
Shift-Tab	Jump to previous track
Alt-Q…I	Jump to track (0…7) MOD N-Channels
Alt-A…K	Jump to track (8…15) MOD N-Channels
F9	Jump to beginning of the pattern
F10	Jump to position ¼ through the pattern
F11	Jump to position halfway through the pattern
F12	Jump to position ¾ through the pattern
The key right of LShift	Enter key-off
Caps-Lock	Enter key-off (Windows only)
1	Enter key-off (OS X only)
Del	Delete note or volume column at cursor
Shift-Del	Delete note, volume and effect at cursor
Ctrl-Del	Delete volume and effect at cursor
Alt-Delete	Delete effect at cursor
Ins	Insert space on current track at cursor position (F13 on mac)
Shift-Ins	Insert row at cursor position (shift-F13 on mac)
Backspace	Delete previous note
Shift-Backspace	Delete previous row

Clipboard operations:

Alt-Cursor keys	Select block
Shift-Alt-Cursor keys	Extend block
Alt-F3	Cut block
Alt-F4	Copy block (yes, even under Windows =)
Alt-F5	Paste block
Alt-F6	Porous paste block
Shift-F3	Cut track
Shift-F4	Copy track
Shift-F5	Paste track
Shift-F6	Porous paste track
Ctrl-F3	Cut pattern
Ctrl-F4	Copy pattern
Ctrl-F5	Paste pattern
Ctrl-F6	Porous paste pattern

Additional shortcuts (not found in FT2):

Ctrl-Alt-Z	Undo
Ctrl-Alt-Y	Redo
Ctrl-Alt-A	Select entire pattern
Shift-I	Interpolate values

Volume scaling:

Alt-V	Volume scale block
Shift-V	Volume scale track
Ctrl-V	Volume scale pattern

Command/Volume macro:

Shift-Alt-1…0	Read command/volume at cursor

Alt-1…0	Write command/volume at cursor

Transpose:

Alt-F7	Transpose current instrument in block down
Alt-F8	Transpose current instrument in block up
Shift-F7	Transpose current instrument in track down
Shift-F8	Transpose current instrument in track up
Ctrl-F7	Transpose current instrument in pattern down
Ctrl-F8	Transpose current instrument in pattern up
Alt-F1	Transpose all instruments in block down
Alt-F2	Transpose all instruments in block up
Shift-F1	Transpose all instruments in track down
Shift-F2	Transpose all instruments in track up
Ctrl-F1	Transpose all instruments in pattern down
Ctrl-F2	Transpose all instruments in pattern up

Instrument selection:

Shift-Up	Select previous instrument
Shift-Down	Select next instrument
Ctrl-Shift-Up	Select previous sample
Ctrl-Shift-Down	Select next sample

You can also quick-type the hex-number of the instrument you want to select on the numeric keypad, the layout is like this:

PC	Mac	
Num 0…9	**Num 0…9**	Digit 0…9
Num /	**Num =**	Digit A
Num *	**Num /**	Digit B
Num -	**Num ***	Digit C
Num +	**Num -**	Digit D
Num Enter	**Num +**	Digit E
Num ,	**Num Enter**	Digit F

Sample editor:

Shift & drag	Quick draw
Ctrl & drag	Resize selection
Alt & drag	Move selection or loop range

6. Effect command reference

I. Glossary

BPM	Traditionally Beats Per Minute, but in tracker terminology it defines the speed of ticks.
Effect memory	When an effect command is called with 0 parameters, previous parameters are used.
Row/line	Refers to one line of "text" on a pattern. In playback its duration depends on how many ticks there are per row (Speed) and fast they are (BPM).
Sample fine-tune/volume/panning	Per sample default settings available through the instrument editor (thus also called instrument volume etc). Overrideable with effect commands. .MODs support these as well but with lower precision. (Save module and load back to enforce .MOD precision.)
Tick	The base time unit in traditional trackers like MilkyTracker, originating from Amiga. Notes are triggered on the first tick of a row (unless delayed) and effects are applied on the following ticks.
Semitone	The smallest musical interval in Western music and in MilkyTracker. A C# note is one semitone away from the note C.
Speed (Spd.)	Number of ticks per row.

II. Effect commands

Standard commands (.MOD &.XM)

- **0xy** Arpeggio
- **1xx** Portamento up
- **2xx** Portamento down
- **3xx** Portamento to note
- **4xy** Vibrato
- **5xy** Portamento to note with volume slide
- **6xy** Vibrato with volume slide
- **7xy** Tremolo
- **8xx** Set note panning position
- **9xx** Sample offset
- **Axy** Volume slide
- **Bxx** Jump to order
- **Cxx** Set note volume
- **Dxx** Pattern break
- **Exy** Subcommands:
  - **E0x** Amiga LED Filter toggle *
  - **E1x** Fine portamento up
  - **E2x** Fine portamento down
  - **E3x** Glissando control **
  - **E4x** Vibrato control **
  - **E5x** Set note fine-tune
  - **E6x** Pattern loop
  - **E7x** Tremolo control **
  - **E8x** Set note panning position ***
  - **E9x** Re-trigger note
  - **EAx** Fine volume slide up
  - **EBx** Fine volume slide down
  - **ECx** Note cut
  - **EDx** Note delay
  - **EEx** Pattern delay
  - **EFx** Funk it! *
- **Fxx** Set song speed/BPM

Extended commands (.XM only)

- **Gxx** Set global volume
- **Hxy** Global volume slide
- **Kxx** Key-off
- **Lxx** Set envelope position
- **Pxy** Panning slide
- **Rxy** Re-trigger note with volume slide
- **Txy** Tremor
- **Xxy** Extra fine portamentos:
  - **X1x** Extra fine portamento up
  - **X2x** Extra fine portamento down

Volume column commands (.XM only)

- **xx** Set note volume
- **+x** Volume slide up
- **-x** Volume slide down
- **Dx** Fine volume slide down (displayed as ▼x)
- **Lx** Panning slide left (displayed as ◀x)
- **Mx** Portamento to note
- **Px** Set note panning position
- **Rx** Panning slide right (displayed as ▶x)
- **Sx** Set vibrato speed
- **Ux** Fine volume slide up (displayed as ▲x)
- **Vx** Vibrato

*) Not implemented, no plans to support
**) Not implemented yet, will be required for feature completeness
***) Not supported on Amiga nor in FT2, effect relocation (8xx, Px) advised

0xy Arpeggio

Syntax: **0**

x = semitone offset

y = semitone offset

Example:

```
C-4 ·1 ·· 037
··· ·· ·· 037
··· ·· ·· 037
··· ·· ·· 037
```

Explanation: Arpeggio quickly alters the note pitch between the base note (C-4) and the semitone offsets **x** (3 = D#4) and **y** (7 = G-4). Each pitch is played for the duration of 1 tick. If speed is higher than 3 (meaning there are more than 3 ticks per row), the sequence is looped.

ProTracker 2/3
Base note is played for tick 0, then the semitone offset **x** for tick 1, then semitone offset **y** for tick 2.

Fasttracker II
Base note is played for tick 0, then the semitone offset **y** for tick 1, then semitone offset **x** for tick 2.

Notes: In MilkyTracker you don't have to and indeed you CAN'T enter the effect digit 0. Just start with the parameter digits and the effect digit will be filled in.

Doesn't have effect memory and cannot be used without parameters.

In Fasttracker II, arpeggio logic fails when song speed is 16 (0x10) or higher. Using arpeggio at such speeds may cause unpredictable results across different players.

Tips: When both effect parameters are used, it is wise to use a song speed value divisible by 3 in order that the arpeggio sequence can loop smoothly.

1xx Portamento up

Syntax: **1**

xx = portamento speed

Example:

```
C-4 ·1 ·· 103
··· ·· ·· 103
··· ·· ·· 103
··· ·· ·· 103
```

Explanation: Portamento is used to slide the note pitch up or down. The higher the **xx**, the faster it goes. Effect is applied on every tick.

Amiga frequencies
The slide speed also depends on the sample frequency.

Notes: ProTracker 2/3
Doesn't have effect memory and cannot be used without parameters.

2xx Portamento down

Syntax: **2**

xx = portamento speed

Example:

```
C-4 ·1 ·· 203
··· ·· ·· 203
··· ·· ·· 203
··· ·· ·· 203
```

Explanation: Works similarly to **1xx** portamento up, only bending note pitch down instead of up.

Notes: ProTracker 2/3
Doesn't have effect memory and cannot be used without parameters.

3xx Portamento to note

Syntax: **3**

xx = portamento speed

Example:

```
C-4 ·1 ·· ···
E-4 ·1 ·· 304
··· ·· ·· 300
··· ·· ·· 310
```

Explanation: This portamento command bends the already playing note pitch towards another one, entered with the **3xx** command. In the example, C-4 is bent towards E-4 at portamento speed **04** which isn't fast enough to reach the E-4 pitch during the two rows at the default song speed (6/125). However, **310** on the following row continues the portamento and being much faster, achieves the target E-4 pitch.

4xy Vibrato

Syntax: **4**
x = speed
y = depth

Example:

```
C-4 ·1 ·· 481
··· ·· ·· 402
··· ·· ·· 400
··· ·· ·· 460
```

Explanation: Vibrato alters note pitch up and down in the maximum range of a full tone. After the initial **xy** pair, parameters can be set individually. The pitch is reset when the command is discontinued.

5xy Portamento to note with volume slide

Syntax: **5**
x = volume slide up speed
y = volume slide down speed

Example:

```
C-4 ·1 ·· ···
E-4 ·1 ·· 304
··· ·· ·· 504
··· ·· ·· 504
```

Explanation: Performs portamento to note with parameters initialized with **3xx** or **Mx** while sliding volume similarly to **Axy volume slide**.

Notes: ProTracker 2/3
Doesn't have effect memory for volume slide speeds, **500** works identically to **300**.

6xy Vibrato with volume slide

Syntax: **6**
x = volume slide up speed
y = volume slide down speed

Example:

```
C-4 ·1 ·· 481
··· ·· ·· 601
··· ·· ·· 600
··· ·· ·· 6C0
```

Explanation: Performs vibrato with parameters initialized with **4xy** or **Sx**+**Vx** while sliding volume similarly to **Axy volume slide**.

Notes: ProTracker 2/3
Doesn't have effect memory for volume slide speeds, **600** works identically to **400**.

7xy Tremolo

Syntax: **7**
x = speed
y = depth

Example:

```
C-4 ·1 ·· 787
··· ·· ·· 700
··· ·· ·· 7C0
··· ·· ·· 700
```

Explanation: Tremolo alters note volume up and down. After the initial **xy** pair, parameters can be set individually. The volume is not reset when the command is discontinued.

8xx Set note panning position

Syntax:	**8** **xx** = panning position

Example:	`C-4 ·1 ·· 880`
	`··· ·· ·· 8A0`
	`··· ·· ·· 8C0`
	`··· ·· ·· 8F0`

Explanation:	Sets the note stereo panning from far left **00** to far right **FF** overriding sample panning setting.
Notes:	ProTracker 2/3 On Amiga, the 4 MOD channels are hard panned left, right, right and left by hardware, no use panning manually there. Fasttracker II Panning envelopes operate relative to the set position.

9xx Sample offset

Syntax:	**9** **xx** = sample offset

Example:	`C-4 ·1 ·· ···`
	`··· ·· ·· ···`
	`C-4 ·1 ·· 908`
	`··· ·· ·· ···`

Explanation:	The sample that the note triggers is played from offset **xx**. The offsets are spread 256 samples apart so **908** skips the first (0x8*256=) 2048 bytes of the sample and plays it on from there. This means that the furthest point **9xx** can reach is (0xFF*256 =) 65280 bytes into the sample.
Tips:	Resampling a loop to exactly (0x10000=) 65536 bytes gives you the highest possible level of control over the sample.

Axy Volume slide

Syntax:	**A** **x** = volume slide up speed **y** = volume slide down speed

Example:	`C-4 ·1 ·· A04`
	`··· ·· ·· A04`
	`C-4 ·1 ·· A0F`
	`··· ·· ·· A80`

Explanation:	Slides note volume up/down at speed **x**/**y** depending on which parameter is specified. Effect is applied per tick so song speed value acts as a multiplier.
Notes:	Parameters **x** and **y** should NOT be used at the same time, doing so almost guarantees unpredictable results across different players. ProTracker 2/3 Doesn't have effect memory and cannot be used without parameters.

Bxx Jump to order

Syntax:	**B** **xx** = song position

Example:	`C-4 ·1 ·· ···`
	`··· ·· ·· ···`
	`··· ·· ·· ···`
	`··· ·· ·· B04`

Explanation:	Immediately breaks the current pattern and jumps to order **xx** in the pattern order table (POT).
Tips:	Can be used to divide a song into separate looping sections effectively creating multiple songs using the same set of instruments. Such modules can be used in games and such where the sections can be triggered dynamically by program events.

Cxx Set note volume

Syntax: **C**
xx = volume

Example:

```
C-4 ·1 ·· ···
··· ·· ·· C10
··· ·· ·· C40
··· ·· ·· C00
```

Explanation: Sets the note volume **00** – **40** overriding sample volume setting.

Notes: Fasttracker II
Volume envelopes operate relative to the set volume.

Dxx Pattern break

Syntax: **D**
xx = row number on next pattern

Example:

```
C-4 ·1 ·· ···
··· ·· ·· ···
··· ·· ·· ···
··· ·· ·· D04
```

Explanation: Breaks the current pattern and jumps to row **xx** on the next pattern.

Notes: Unlike with the majority of effect parameters, here **xx** is a decimal value rather than hexadecimal. Hexadecimal values are accepted but the first digit is still interpreted as decimal so it's best to avoid hex this time.

The highest row number you can jump to is 63.

E1x Fine portamento up

Syntax: **E1**
x = portamento speed

Example:

```
C-4 ·1 ·· E11
··· ·· ·· E12
··· ·· ·· E13
··· ·· ·· E14
```

Explanation: Works similarly to **1xx portamento up**, only the slide is a lot finer because the effect is applied only once per row.

E2x Fine portamento down

Syntax: **E2**
x = portamento speed

Example:

```
C-4 ·1 ·· E11
··· ·· ·· E12
··· ·· ·· E13
··· ·· ·· E14
```

Explanation: Works similarly to **2xx portamento down** bending note pitch down, only the slide is a lot finer like with **E1x**.

E3x Glissando control

Syntax: **E3**
x = glissando control toggle on/off

Example:

```
C-4 ·1 ·· E31
D-4 01 ·· 305
··· ·· ·· 300
··· ·· ·· E30
```

Explanation: Glissando control **E31** changes note portamento behavior affecting commands **3xx**, **5xy** and **Mx**. Instead of stepless pitch bend (=glissando), the frequencies are rounded to nearest semitone. To revert to default glissando, use **E30**.

Notes: This command is not yet implemented in MilkyTracker.

E4x Vibrato control

Syntax: **E4**
x = vibrato waveform selection

Example:

```
C-4 ·1 ·· 48C
··· ·· V0 E41
··· ·· V0 E42
··· ·· ·· E40
```

Explanation: This command sets the waveform used for **4xy**, **6xy** and **Vx** vibrato commands. The default waveform is sine, reset on every new note (**E40**). Possible parameter **x** values are:
0 = Sine
1 = Ramp down
2 = Square
4 = Continuous sine
5 = Continuous ramp down
6 = Continuous square

Notes: This command is not yet implemented in MilkyTracker.

E5x Set note fine-tune

Syntax: **E5**
x = fine-tune

Example:

```
C-4 ·1 ·· E54
··· ·· ·· ···
C-4 ·1 ·· E5C
··· ·· ·· ···
```

Explanation: Sets note fine-tune overriding sample fine-tune setting. This command works a little differently for .MOD and .XM tracking. While both parameter value ranges are logical, the latter is also linear. See here:

x	ProTracker 2/3	Fasttracker II
0	0	-128
1	+16	-112
2	+32	-96
3	+48	-80
4	+64	-64
5	+80	-48
6	+96	-32
7	+112	-16
8	-128	0
9	-112	+16
A	-96	+32
B	-80	+48
C	-64	+64
D	-48	+80
E	-32	+96
F	-16	+112

E6x Pattern loop

Syntax: **E6**
x = set loop point / number of iterations

Example:

```
C-4 ·1 ·· E60
··· ·· ·· ···
F-4 01 ·· ···
··· ·· ·· E63
```

Explanation: Loops a section of a pattern **x** times. **E60** sets the (optional) loop start point and **E6x** with **x** values **1**–**F** sets the end point and the number of iterations. If loop start point is not set, beginning of the pattern is used by default.

Notes: The loop points need to be set on the same channel for them to work correctly.

Fasttracker II

One of the most (in)famous FT2 bugs is the **`E60`** bug: When **`E60`** is used on a pattern row *x*, the following pattern also starts from row *x* instead of the beginning of the pattern. This can be avoided by placing a **`D00` pattern break** on the last row of the pattern where **`E60`** was used.

Tips: Musicians concerned with correct playback of their .XM modules can utilize the **`E60`** bug to skip sections of (or the whole) song when played with lesser players. `;`)

E7x Tremolo control

Syntax: **E7**

x = tremolo waveform selection

Example:

```
C-4 ·1 ·· E72
··· ·· ·· 76C
··· ·· ·· E70
··· ·· ·· 700
```

Explanation: This command sets the waveform used for **`7xy` tremolo** command. As with **`E4x` vibrato control**, the default waveform is sine and the possible parameter **`x`** values are:

`0` = Sine
`1` = Ramp down
`2` = Square
`4` = Continuous sine
`5` = Continuous ramp down
`6` = Continuous square

Notes: This command is not yet implemented in MilkyTracker.

E8x Set note panning position

Syntax: **E8**

x = panning position

Explanation: This command is another panning position command used by some trackers…

Notes: …However, since it does not work on Amiga (because of the hardware panning) nor in Fasttracker II (hmm, enough panning commands already?), effect relocation to **`8xx`** or **`Px`** is advised in order to produce compatible modules.

E9x Re-trigger note

Syntax: **E9**

x = triggering interval

Example:

```
C-4 ·1 ·· E93
C-4 ·1 ·· ···
··· ·· ·· ···
C-4 ·1 ·· ···
```

Explanation: This command re-triggers a note every **`x`** ticks.

EAx Fine volume slide up

Syntax: **EA**

x = speed

Example:

```
C-4 ·1 10 EA2
··· ·· ·· EA0
··· ·· ·· EA4
··· ·· ·· EA0
```

Explanation: Works similarly to **`Ax0` volume slide** up, only the slide is a lot finer because the effect is applied only once per row.

EBx Fine volume slide down

Syntax: **EB**

x = speed

Example:

```
C-4 ·1 ·· EB2
```

```
··· ·· ·· EB0
··· ·· ·· EB4
··· ·· ·· EB0
```

Explanation: Works similarly to **`A0y` volume slide** down, only the slide is a lot finer like with **`EAx`**.

ECx Note cut

Syntax: **`EC`**

x = tick number

Example:

```
C-4 ·1 ·· EC1
C-4 ·1 ·· EC2
C-4 ·1 ·· ···
··· ·· ·· EC0
```

Explanation: Cuts a note by setting its volume to 0 at tick precision. Possible parameter **x** values are **`0`** – (song speed - 1). Higher values have no effect.

EDx Note delay

Syntax: **`ED`**

x = tick number

Example:

```
C-4 ·1 ·· ···
A#3 01 ·· ED3
C-4 ·1 ·· ···
··· ·· ·· ···
```

Explanation: Delays a note **x** ticks. Like with **`ECx` note cut**, possible **x** values are **`0`** – (song speed - 1). Higher values prevent the note from playing altogether.

EEx Pattern delay

Syntax: **`EE`**

x = amount of rows

Example:

```
C-4 ·1 ·· ···
A#3 01 ·· EE5
C-4 ·1 ·· ···
··· ·· ·· ···
```

Explanation: Delays playback progression for the duration of **x** rows.

Fxx Set song speed/BPM

Syntax: **`F`**

xx = speed/BPM value

Example:

```
C-4 ·1 ·· F90
A#3 01 ·· F03
C-4 ·1 ·· ···
··· ·· ·· ···
```

Explanation: Parameter **x** values **`01`** – **`1F`** set song speed i.e. the amount of ticks per row. Values **`20`** – **`FF`** set the BPM which essentially is the speed of the ticks. **`F00`** stops playback.

Gxx Set global volume

Syntax: **`G`**

xx = volume

Example:

```
C-4 ·1 ·· G40
··· ·· ·· G20
··· ·· ·· G10
··· ·· ·· G00
```

Explanation: Sets the global song note volume **`00`** – **`40`**.

Hxy Global volume slide

Syntax: **`H`**
x = volume slide up speed
y = volume slide down speed

Example:

```
C-4 ·1 ·· H04
··· ·· ·· H04
C-4 ·1 ·· H0F
··· ·· ·· H80
```

Explanation: Slides global song volume up/down at speed **x**/**y** depending on which parameter is specified.

Notes: Parameters **x** and **y** should NOT be used at the same time, doing so almost guarantees unpredictable results across different players.

Kxx Key-off

Syntax: **`K`**
xx = tick number

Example:

```
C-4 ·1 ·· K03
··· ·· ·· ···
C-4 ·1 ·· ···
··· ·· ·· K00
```

Explanation: Sends instrument key-off much like the note column counterpart, only in tick precision. As **`K00`** is the equivalent of a note column key-off, it cancels any actual note on the row. Possible parameter **xx** values are **`00`** – (song speed - 1). Higher values have no effect.

Lxx Set envelope position

Syntax: **`L`**
xx = envelope position

Example:

```
C-4 ·1 ·· L20
··· ·· ·· ···
··· ·· ·· L00
··· ·· ·· ···
```

Explanation: Makes the currently playing note jump to tick **xx** on the volume envelope timeline.
If the **volume** envelope's sustain point is set, the **panning** envelope also jumps to tick **xx** (This is an original FT2 quirk).

Pxy Panning slide

Syntax: **`P`**
x = panning slide right speed
y = panning slide left speed

Example:

```
C-4 ·1 ·· P04
··· ·· ·· P00
··· ·· ·· P80
··· ·· ·· P00
```

Explanation: Slides note panning right/left at speed **x**/**y** depending on which parameter is specified. Effect is applied per tick so song speed acts as a multiplier.

Notes: Parameters **x** and **y** should NOT be used at the same time, doing so almost guarantees unpredictable results across different players.

Rxy Re-trigger note with volume slide

Syntax: **`R`**
x = volume slide speed
y = triggering interval

Example:

```
C-4 ·1 ·· R81
··· ·· ·· R12
··· ·· ·· R23
··· ·· ·· R04
```

Explanation: Much like **E9x**, this command rapidly re-triggers a note while sliding its volume. Parameter **x** values affect note volume like this:
0 = previous **x** value
1 = - 1
2 = - 2
3 = - 4
4 = - 8
5 = -16
6 = * 0.66666666666666666666666666666667
7 = * 0.5
8 = no volume change
9 = + 1
A = + 2
B = + 4
C = + 8
D = +16
E = * 1.5
F = * 2

Notes: This command is very buggy from the start, straight from the source, Fasttracker II. While FT2's own documentation is inaccurate in many places, this is different. Extensive testing has revealed almost bizarre qualities of this effect and it's up to MilkyTracker to emulate it all. Without doubt the quirk the team has spent the most time and iterations working on getting it right. And still we advise to be careful with it. When using **Rxy**, check your song with FT2 (render to .WAV if you don't have the hardware (to emulate)), or at least BASS/XMPlay. And if you do find something odd, please report the bug as accurately and detailed as possible.

Setting volume on the volume column (**xx**) at the same time with **Rxy** resets the volume to **xx** before each re-trigger making the effect sound different.

Tips Use **R8y** instead of **R0y** when you want to keep the volume unchanged, these two **x** values are often documented inaccurately as "No volume change" and "Unused", respectively.

Txy Tremor

Syntax: **T**
x + 1 = ticks on
y + 1 = ticks off

Example:

```
C-4 ·1 ·· T13
··· ·· ·· T00
··· ·· ·· T31
··· ·· ·· T00
```

Explanation: Rapidly alters note volume from full to zero, **x** + 1 and **y** + 1 setting the duration of the states in ticks.

Notes: Using **T00** makes a fast tremor effect but it also functions as effect memory repeating the last parameters. So you can only use this **00** speed once per channel before you use any other parameter values.

X1x Extra fine portamento up

Syntax: **X1**
x = speed

Example:

```
C-4 ·1 ·· X11
··· ·· ·· X10
··· ·· ·· X18
··· ·· ·· X10
```

Explanation: Works just like **E1x fine portamento up**, only with 4 times the precision.

X2x Extra fine portamento down

Syntax: **X2**
x = speed

Example:

```
C-4 ·1 ·· X11
··· ·· ·· X20
··· ·· ·· X28
··· ·· ·· X20
```

Explanation: Works just like **E2x fine portamento down**, only with 4 times the precision like **E1x extra fine portamento up**.

xx Set note volume

Syntax: **xx** = volume

Example:

```
C-4 ·1 ·· ···
··· ·· 10 ···
··· ·· 40 ···
··· ·· 00 ···
```

Explanation: Sets the note volume **00** – **40** overriding sample volume setting. This is what the volume column is primarily used for, hence no effect command character. It's the equivalent of **Cxx set note volume** on the effect column.

+x Volume slide up

Syntax: **+**
x = speed

Example:

```
C-4 ·1 10 ···
··· ·· +2 ···
··· ·· +4 ···
··· ·· +8 ···
```

Explanation: Slides note volume up at speed **x** with the song speed (ticks) acting as a multiplier like with **Ax0** volume slide on the effect column.

-x Volume slide down

Syntax: **-**
x = speed

Example:

```
C-4 ·1 ·· ···
··· ·· -2 ···
··· ·· -4 ···
··· ·· -8 ···
```

Explanation: Slides note volume down at speed **x** with the song speed (ticks) acting as a multiplier like with **A0y** volume slide on the effect column.

Dx Fine volume slide down (displayed as ▼x)

Syntax: **D**
x = speed

Example:

```
C-4 ·1 ·· ···
··· ·· ▼2 ···
··· ·· ▼4 ···
··· ·· ▼8 ···
```

Explanation: This is the volume column equivalent of **EBx fine volume slide down**, effect is applied once per row.

Lx Panning slide left (displayed as ◀x)

Syntax: **L**
x = speed

Example:

```
C-4 ·1 ·· ···
··· ·· ◀2 ···
··· ·· ◀4 ···
··· ·· ◀8 ···
```

Explanation: Slides note panning left at speed **x** with the song speed value (ticks) acting as a multiplier like with **P0x volume slide** on the effect column.

Mx Portamento to note

Syntax: **M**
x = speed

Example:

```
C-4 ·1 ·· ···
E-4 01 ·· 304
··· ·· M0 ···
··· ·· M1 ···
```

Explanation: This is the volume column equivalent of **3xx portamento**, only with 1 digit resolution. **M1** corresponds to **311**, **M2** to **322** and so on…

Tips: **3xx** and **Mx** share effect memory, so it's possible to initialize a portamento with a more precise **3xx** value and sustain it with **M0** freeing the effect column for arpeggios, note delays, tremolo or whatever.

Px Set note panning position

Syntax: **P**
x = speed

Example:

```
C-4 ·1 P4 ···
··· ·· ·· ···
··· ·· PC ···
··· ·· ·· ···
```

Explanation: This is the volume column equivalent of **8xx panning**, only with 1 digit resolution. **P8** corresponds to **888**, **P9** to **899** and so on…

Tips: **3xx** and **Mx** share effect memory, so it's possible to initialize a portamento with a more precise **3xx** value and sustain it with **M0** freeing the effect column for arpeggios, note delays, tremolo or whatever.

Rx Panning slide right (displayed as ▶x)

Syntax: **R**
x = speed

Example:

```
C-4 ·1 ·· ···
··· ·· ▶2 ···
··· ·· ▶4 ···
··· ·· ▶8 ···
```

Explanation: Slides note panning right at speed **x** with the song speed value (ticks) acting as a multiplier like with **Px0 volume slide** on the effect column.

Sx Set vibrato speed

Syntax: **S**
x = speed

Example:

```
C-4 ·1 ·· 48F
··· ·· S4 A01
··· ·· ·· 600
··· ·· ·· 400
```

Explanation: Sets the vibrato speed like the **x** in **4xy vibrato**. In the example it is used instead of **4xy** to free up the effect column.

Ux Fine volume slide up (displayed as ▲x)

Syntax: **U**
x = speed

Example:

```
C-4 ·1 10 ···
··· ·· ▲2 ···
··· ·· ▲4 ···
··· ·· ▲8 ···
```

Explanation: This is the volume column equivalent of **EAx fine volume slide up**, effect is applied once per row.